Final Technical Report for NASA Research Grant NAG-1-676
Design of Adaptable Distributed Systems

Principal Investigator: B. Bhargava
Department of Computer Sciences
Purdue University
West Lafayette, In 47907
(317) 494-6013
bb@cs.purdue.edu

Research Assistants: A. Helal, C. Koelbel,
S. Lian, E. Mafla, J. Mullen, J. Riedl,
J. Srinivasan, J. Sang

The focus of our research is the investigation of the principles necessary to build high performance, reliable, and adaptable distributed systems. We have conducted both theoretical and experimental studies in the areas of replicated copy control, site failure/recovery, network partition, concurrent checkpointing, and communication systems. We have studied the relationship between database and operating system.

Our research has applications in the embedded distributed system that is being designed for NASA's space station. Our work in O-raid has applications in the management of complex databases and information that is received during experiments in space.

This research contract was awarded jointly with Army Institute for Research in Management, Information, and Computer Sciences (AIRMICS). Our results in the design of Raid system are applicable to the goals of interoperability of systems being managed for corporate databases.

During the conduct of this research, the principal investigator and his students made three presentations to NASA, four to AIRMICS (attended by NASA staff), and four presentations at Purdue University. This research has resulted in over twenty publications in International Journals and Conferences. The results have been made widely available to researchers of NASA and ARMY as well as to many researchers in academia and industry. AT&T and Unisys have supported this work through grants/gifts to Purdue University.

We now outline the various theoretical, implementation, and experimental aspects of this work. We include with this report a selected list of research papers.

Theoretical Studies:

We have developed a formal model for adaptability in transaction processing algorithms and identified the correctness conditions for switching algorithms at run-time [2]. This model has been used for concurrency control, replication control, distributed commitment, and network partition algorithms. We have designed quorum-based replication control methods that are adaptable to the type of object representation and to different kinds of failures. The quorum model, when extended to allow dynamic changes to quorum assignments, provides a generic framework that encompasses a wide variety of replication control methods, including token passing, available copies, and

dynamic voting. Types of object representation range from value-based, in which an object is stored as a single entity and a write operation overwrites the value of this entity, to event-based, in which an object is stored as a history of timestamped operations and a write operation appends to this list.

The research in the study of concurrent checkpointing algorithms for distributed systems has resulted in a formal model using transactions [8], two classes of algorithms [9,10], and experimental work [11]. Our algorithms allows concurrent checkpointing as well as recovery even in the presence of failures of process and network partitions. One class of algorithm is based on the coordinated approach [9] while the algorithm based on the second class allows independent checkpointing and concurrent rollback and recovery while the non-faulty processes continue their execution optimistically [10].

Our latest research effort has been directed towards developing an object-relation model for complex and emerging database applications in business, and space applications. The query language SQL has been extended to SQL++ to deal with objects in addition to relations [12,13]. This research resulted in the design of O-raid (Object-oriented RAID).

Experimental Prototype RAID

We have built a prototype called RAID (Robust and Adaptable Distributed Database Systems) [1] that runs in our laboratory. The implementation has been carried out on a network of SUN workstations. The system is a collection of autonomous servers connected by a communication system [4]. Each site runs servers that implement access management with stable storage, transaction execution, concurrency control, replicated copy management, and other services for transaction processing. In order to provide infrastructure for experiments in adaptability and replication control, a new version of the Raid software with a new control flow among servers has been developed. In addition we have implemented a mini-Raid system that allows experimentation with a variety of protocols. Mini-Raid system has been used for experiments in site failure algorithms for both fully replicated database environment as well as partial replication [7]. In addition, it has been used for measurements in distributed checkpointing.

The Raid communication system can run either UDP/IP datagrams or a high speed ethernet protocol. High level calls exist for services such as reliable multicast necessary for distributed transaction commitment. A name server has been implemented to provide location independent addressing for various servers. This server keeps track of failure/recovery of sites and the integration of new sites to the system.

We have identified the problems that arise in general purpose interprocess communication mechanisms available for the Raid distributed database transaction processing system by conducting a series of experiments [5,6]. These mechanisms are CPU-intensive and optimized only for remote communication and do not support multicasting. We have developed an advanced set of services that include multicast, remote procedure calls (RPC), inexpensive datagram transmission, and efficient local interprocess communication (IPC).

We have developed a transaction-oriented communication facility [6]. Its performance is limited only by the network device driver and the system call mechanism

overheads. Sending a 100-byte message (monocast or multicast) takes 650 microseconds, which includes the overheads. This is approximately 30% of the cost of the corresponding Unix communication facility. Our communication facility demonstrates the feasibility of address spaces for structuring complex distributed transaction processing systems. It employs shared-memory ports, a simple naming scheme, and a transaction-oriented multicasting mechanism. Local and remote communication is through ports. Ports can be accessed directly by the kernel and by user-level processes. The naming scheme used for the application and network levels avoids the use of name-resolution protocols by directly mappings the application-level name space to the network name space. Physical multicasting is used and the need for special protocols to agree on a group address is avoided. Each transaction defines a multicasting group consisting of the set of sites involved. A group's multicasting address is a function of the corresponding transaction identifier and can be independently determined by each member of the group. The new communication facility reduces kernel overhead during transaction processing in Raid by up to 70%.

Experimental Studies

Our experimental effort is currently focussed on communication software. We have developed two software systems for this effort: RAID system [1], and the PUSH system [3].

The Push system has been developed to make measurements on transaction processing services and communication facilities that run at the operating system kernel. Further studies that explore the operating system support for database systems are underway and will lead to the design of a real-time processing system.

As stated in the previous section, we have worked extensively in developing a variety of communication services. We have conducted extensive experimentation [4,5,6] that measure communication performance in five areas: (i) layered implementations of communication protocols, (ii) special-purpose local interprocess communication methods, (iii) a kernel--level hardware-independent multicasting facility, (iv) the Raid system running on different network configurations, and (v) Push, a facility for extending kernel services. Push allows us to conduct measurements by supplementing and/or modifying the communication facilities in the operating system kernel while it is running.

References

[1] B. Bhargava, J. Riedl, "RAID Distributed Database System", *IEEE Transactions in Software Engineering*, June, 1989.

[2] B. Bhargava, J. Riedl, "A Formal Model for Adaptable Systems for Transaction Processing", *IEEE Transactions on Knowledge and Data Engineering*, Dec., 1989.

[3] B. Bhargava, E. Mafla, J. Riedl, "Experimental Facility for Kernel Extensions to Support Distributed Database Systems", Department of Computer Science,

Technical Report Number CSD-TR-930, Purdue University, Oct., 1989.

[4] B. Bhargava, E. Mafla and J. Riedl, "Communication in the Raid Distributed Database System", in *Proceedings of International Phoenix Conference on Computers and Communications*, March, 1990. (Full version in Journal of Computer Networks and ISDN Systems, Feb., 1991).

[5] B. Bhargava, E. Mafla, J. Riedl, B. Sauder, "Implementation and Measurements of Efficient Communication Facilities for Distributed Database Systems", in *Proceedings of the 5th IEEE Data Engineering Conference*, (Feb., 1989).

[6] E. Mafla and B. Bhargava, "Implementation and Performance of a Communication Facility for Distributed Transaction Processing", in *Proceedings of the Second Usenix Symposium on Experiences with Distributed and Multiprocessor Systems*, Atlanta, Mar., 1991

[7] B. Bhargava, P. Noll, D. Sabo, "An Experimental Analysis of Replicated Copy Control During Site Failure and Recovery", in *Proceedings of the Fourth IEEE Data Engineering Conference*, (February 1988): 82-91.

[8] P. Leu, B. Bhargava, "A Model for Concurrent Checkpointing and Recovery Using Transactions", in *Proceedings of the 9th IEEE Distributed Computing Systems Conference*, (June 1989).

[9] B. Bhargava, P. Leu, "Concurrent Robust Checkpointing and Recovery in Distributed Systems", *Proceedings of the Fourth IEEE Data Engineering Conference*, (February 1988): 154-163.

[10] B. Bhargava, S. Lian, "Independent Checkpointing and Concurrent Rollback for Recovery in Distributed System", in *Proceedings of Seventh IEEE Symposium on Reliability in Distributed Systems*, (1988): 3-12.

[11] B. Bhargava, P. Leu, S. Lian, "Experimental Evaluation of Concurrent Checkpointing and Rollback-Recovery Algorithms", in *Proceedings of 6th IEEE Data Engineering Conference*, Los Angeles, February 1990.

[12] B. Bhargava, P. Dewan, J. Mullen and A. Vikram, "Implementing Object Support in the RAID Distributed Database System", in *Proceedings of First IEEE International Conference on System Integration, Morris Town, NJ., April, 1990.*

[13] P. Dewan, A. Vikram, and B. Bhargava, "Engineering the Object-Relation Database Model in O-Raid", in *Proceedings of the International Conference on Foundations of Data Organization and Theory*, June 1989, Paris